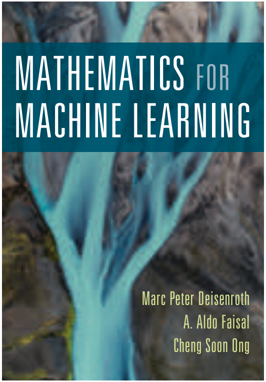
Linear Regression

Marc Deisenroth
Centre for Artificial Intelligence
Department of Computer Science
University College London

@mpd37
m.deisenroth@ucl.ac.uk
https://deisenroth.cc

AIMS Rwanda and AIMS Ghana

March/April 2020



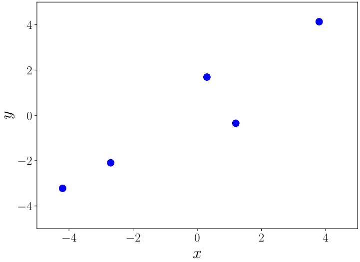
https://mml-book.com

Chapter 9

Regression Problems

Regression (curve fitting)

Given inputs $x \in \mathbb{R}^D$ and corresponding observations $y \in \mathbb{R}$ find a function f that models the relationship between x and y.

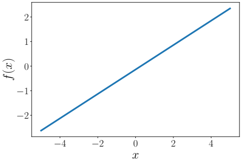


- Typically parametrize the function f with parameters $\boldsymbol{\theta}$
- Linear regression: Consider functions f that are **linear in the parameters**

Linear Regression Functions

- Straight lines

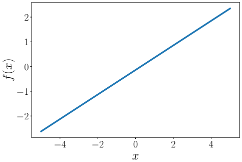
$$y = f(x, \boldsymbol{\theta}) = \theta_0 + \theta_1 x = \begin{bmatrix} \theta_0 & \theta_1 \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix}$$



Linear Regression Functions

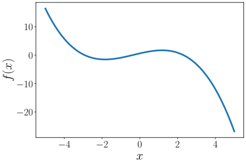
- Straight lines

$$y = f(x, \boldsymbol{\theta}) = \theta_0 + \theta_1 x = \begin{bmatrix} \theta_0 & \theta_1 \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix}$$



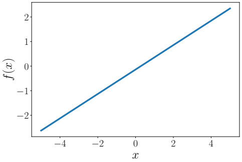
- Polynomials

$$y = f(x, \boldsymbol{\theta}) = \sum_{m=0}^{M} \theta_m x^m = \begin{bmatrix} \theta_0 & \cdots & \theta_M \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ x^M \end{bmatrix}$$



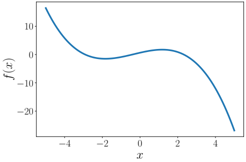
- Straight lines

$$y = f(x, \boldsymbol{\theta}) = \theta_0 + \theta_1 x = \begin{bmatrix}\theta_0 & \theta_1\end{bmatrix}\begin{bmatrix}1 \\ x\end{bmatrix}$$



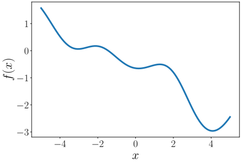
- Polynomials

$$y = f(x, \boldsymbol{\theta}) = \sum_{m=0}^{M} \theta_m x^m = \begin{bmatrix}\theta_0 & \cdots & \theta_M\end{bmatrix}\begin{bmatrix}1 \\ \vdots \\ x^M\end{bmatrix}$$



- Radial basis function networks

$$y = f(x, \boldsymbol{\theta}) = \sum_{m=1}^{M} \theta_m \exp\Big(-\tfrac{1}{2}(x - \mu_m)^2\Big)$$



Linear Regression Model and Setting

$$y = \boldsymbol{x}^\top \boldsymbol{\theta} + \epsilon\,, \quad \epsilon \sim \mathcal{N}\big(0,\,\sigma^2\big)$$

- Given a training set $(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_N, y_N)$ we seek optimal parameters $\boldsymbol{\theta}^*$

Linear Regression Model and Setting

$$y = \boldsymbol{x}^\top \boldsymbol{\theta} + \epsilon\,, \quad \epsilon \sim \mathcal{N}\big(0,\, \sigma^2\big)$$

- Given a training set $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)$ we seek optimal parameters $\boldsymbol{\theta}^*$
  - **Maximum Likelihood Estimation**
  - **Maximum a Posteriori Estimation**

Maximum Likelihood

- Define $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_N]^\top \in \mathbb{R}^{N \times D}$ and $\boldsymbol{y} = [y_1, \dots, y_N]^\top \in \mathbb{R}^N$
- Find parameters $\boldsymbol{\theta}^*$ that maximize the likelihood

Maximum Likelihood

- Define $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]^\top \in \mathbb{R}^{N \times D}$ and $\boldsymbol{y} = [y_1, \ldots, y_N]^\top \in \mathbb{R}^N$
- Find parameters $\boldsymbol{\theta}^*$ that maximize the likelihood

$$p(y_1, \ldots, y_N | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N, \boldsymbol{\theta}) = p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) = \prod_{n=1}^{N} \mathcal{N}\big(y_n \,|\, \boldsymbol{x}_n^\top \boldsymbol{\theta},\, \sigma^2\big)$$

Maximum Likelihood

- Define $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]^\top \in \mathbb{R}^{N \times D}$ and $\boldsymbol{y} = [y_1, \ldots, y_N]^\top \in \mathbb{R}^N$
- Find parameters $\boldsymbol{\theta}^*$ that maximize the likelihood

$$p(y_1, \ldots, y_N | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N, \boldsymbol{\theta}) = p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) = \prod_{n=1}^{N} \mathcal{N}\big(y_n \,|\, \boldsymbol{x}_n^\top \boldsymbol{\theta},\, \sigma^2\big)$$

- Log-transformation ▶▶ **Maximize the log likelihood**

Maximum Likelihood

- Define $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]^\top \in \mathbb{R}^{N \times D}$ and $\boldsymbol{y} = [y_1, \ldots, y_N]^\top \in \mathbb{R}^N$
- Find parameters $\boldsymbol{\theta}^*$ that maximize the likelihood

$$p(y_1, \ldots, y_N | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N, \boldsymbol{\theta}) = p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) = \prod_{n=1}^{N} \mathcal{N}\big(y_n \,|\, \boldsymbol{x}_n^\top \boldsymbol{\theta},\, \sigma^2\big)$$

- Log-transformation ▶▶ **Maximize the log likelihood**

$$\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) = \sum_{n=1}^{N} \log \mathcal{N}\big(y_n \,|\, \boldsymbol{x}_n^\top \boldsymbol{\theta},\, \sigma^2\big)\,,$$

$$\log \mathcal{N}\big(y_n \,|\, \boldsymbol{x}_n^\top \boldsymbol{\theta},\, \sigma^2\big) = -\frac{1}{2\sigma^2}(y_n - \boldsymbol{x}_n^\top \boldsymbol{\theta})^2 + \text{ const}$$

Maximum Likelihood (2)

With

$$\log \mathcal{N}\big(y_n \,|\, \boldsymbol{x}_n^\top \boldsymbol{\theta},\, \sigma^2\big) = -\frac{1}{2\sigma^2}(y_n - \boldsymbol{x}_n^\top \boldsymbol{\theta})^2 + \text{const}$$

we get

$$\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) = \sum_{n=1}^{N} \log \mathcal{N}\big(y_n \,|\, \boldsymbol{x}_n^\top \boldsymbol{\theta},\, \sigma^2\big) = -\frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \boldsymbol{x}_n^\top \boldsymbol{\theta})^2 + \text{const}$$

Maximum Likelihood (2)

With

$$\log \mathcal{N}\big(y_n \,|\, \boldsymbol{x}_n^\top \boldsymbol{\theta},\, \sigma^2\big) = -\frac{1}{2\sigma^2}(y_n - \boldsymbol{x}_n^\top \boldsymbol{\theta})^2 + \text{const}$$

we get

$$\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) = \sum_{n=1}^{N} \log \mathcal{N}\big(y_n \,|\, \boldsymbol{x}_n^\top \boldsymbol{\theta},\, \sigma^2\big) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n - \boldsymbol{x}_n^\top \boldsymbol{\theta})^2 + \text{const}$$

$$= -\frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}) + \text{const}$$

Maximum Likelihood (2)

With

$$\log \mathcal{N}\big(y_n \,|\, \boldsymbol{x}_n^\top \boldsymbol{\theta},\, \sigma^2\big) = -\frac{1}{2\sigma^2}(y_n - \boldsymbol{x}_n^\top \boldsymbol{\theta})^2 + \text{const}$$

we get

$$\begin{aligned}
\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) &= \sum_{n=1}^{N} \log \mathcal{N}\big(y_n \,|\, \boldsymbol{x}_n^\top \boldsymbol{\theta},\, \sigma^2\big) = -\frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \boldsymbol{x}_n^\top \boldsymbol{\theta})^2 + \text{const} \\
&= -\frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}) + \text{const} \\
&= -\frac{1}{2\sigma^2} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\|^2 + \text{const}
\end{aligned}$$

Maximum Likelihood (2)

With

$$\log \mathcal{N}\big(y_n \,|\, \boldsymbol{x}_n^\top \boldsymbol{\theta},\, \sigma^2\big) = -\frac{1}{2\sigma^2}(y_n - \boldsymbol{x}_n^\top \boldsymbol{\theta})^2 + \text{const}$$

we get

$$\begin{aligned}\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) &= \sum_{n=1}^{N} \log \mathcal{N}\big(y_n \,|\, \boldsymbol{x}_n^\top \boldsymbol{\theta},\, \sigma^2\big) = -\frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \boldsymbol{x}_n^\top \boldsymbol{\theta})^2 + \text{const} \\ &= -\frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}) + \text{const} \\ &= -\frac{1}{2\sigma^2}\,\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\|^2 + \text{const}\end{aligned}$$

- Computing the gradient with respect to $\boldsymbol{\theta}$ and setting it to $\boldsymbol{0}$ gives the **maximum likelihood estimator** (least-squares estimator)

$$\boxed{\boldsymbol{\theta}^{\text{ML}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}}$$

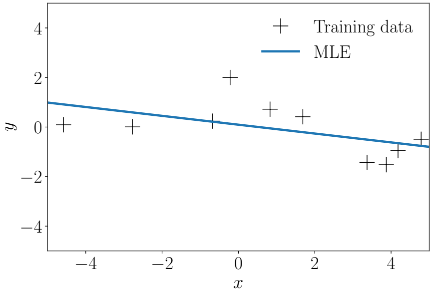
Making Predictions

$$y = \boldsymbol{x}^\top \boldsymbol{\theta} + \epsilon, \quad \epsilon \sim \mathcal{N}\big(0,\, \sigma^2\big)$$

Given an arbitrary input $\boldsymbol{x}_*$, we can predict the corresponding observation y_* using the maximum likelihood parameter:

$$p(y_* | \boldsymbol{x}_*, \boldsymbol{\theta}^{\mathsf{ML}}) = \mathcal{N}\big(y_* \,|\, \boldsymbol{x}_*^\top \boldsymbol{\theta}^{\mathsf{ML}},\, \sigma^2\big)$$

- Measurement noise variance σ^2 assumed known
- In the absence of noise ($\sigma^2 = 0$), the prediction will be deterministic



$$y = \theta_0 + \theta_1 x + \epsilon, \quad \epsilon \sim \mathcal{N}\big(0,\, \sigma^2\big)$$

- At any query point x_* we obtain the mean prediction as

$$\mathbb{E}[y_* | \boldsymbol{\theta}^{\mathsf{ML}}, x_*] = \theta_0^{\mathsf{ML}} + \theta_1^{\mathsf{ML}} x_*$$

Nonlinear Functions

$$y = \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\theta} + \epsilon = \sum_{m=0}^{M} \theta_m x^m + \epsilon$$

- Polynomial regression with features

$$\boldsymbol{\phi}(x) = [1, x, x^2, \ldots, x^M]^\top$$

- Maximum likelihood estimator:

Nonlinear Functions

$$y = \phi(\boldsymbol{x})^\top \boldsymbol{\theta} + \epsilon = \sum_{m=0}^{M} \theta_m x^m + \epsilon$$

- Polynomial regression with features

$$\phi(x) = [1, x, x^2, \dots, x^M]^\top$$

- Maximum likelihood estimator:

$$\boldsymbol{\theta}^{\mathrm{ML}} = (\boldsymbol{\Phi}^\top \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^\top \boldsymbol{y}$$

Example 2: Polynomial Regression

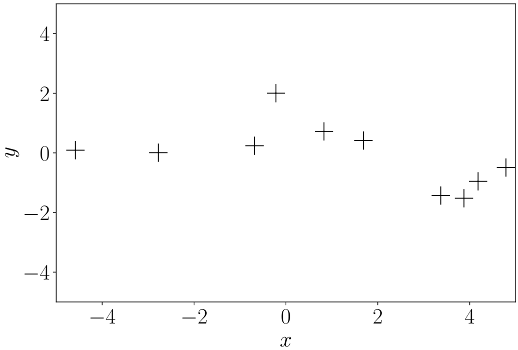


Figure: Training data

Example 2: Polynomial Regression

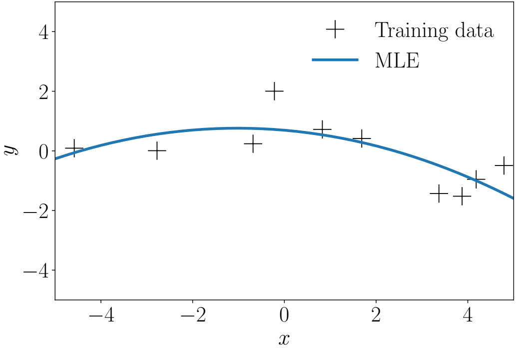


Figure: 2nd-order polynomial

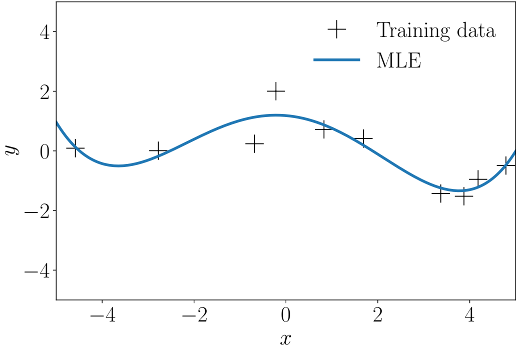


Figure: 4th-order polynomial

Example 2: Polynomial Regression

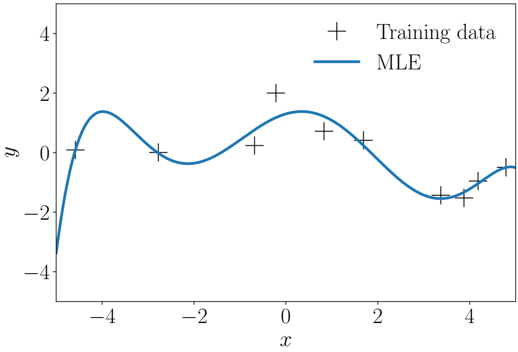


Figure: 6th-order polynomial

Example 2: Polynomial Regression

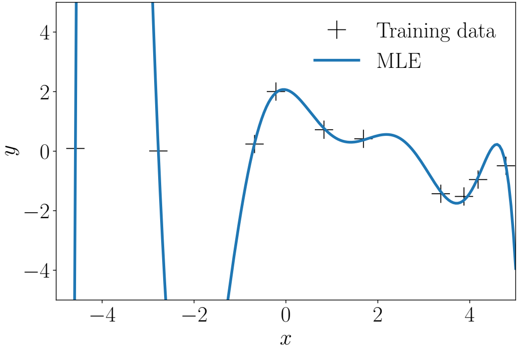


Figure: 8th-order polynomial

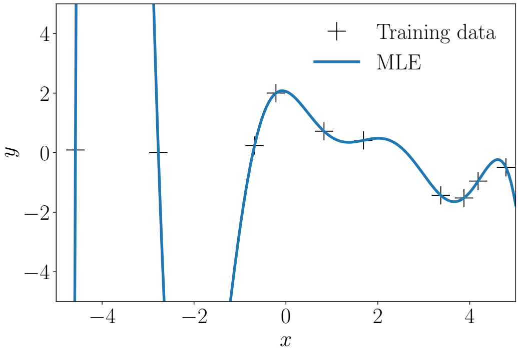
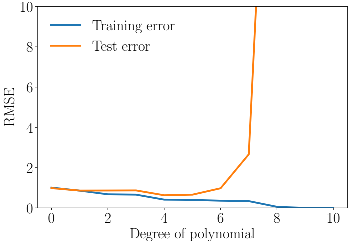


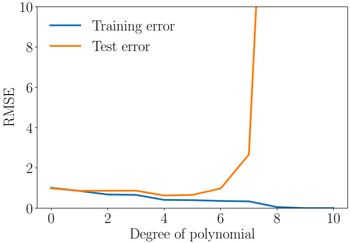
Figure: 10th-order polynomial

Overfitting



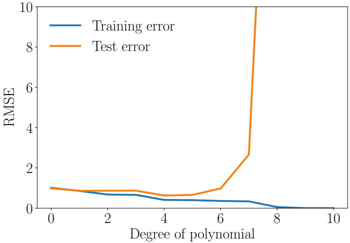
- Training error decreases with higher flexibility of the model

Overfitting



- Training error decreases with higher flexibility of the model
- We are not so much interested in the training error, but in the **generalization error**: How well does the model perform when we predict at previously unseen input locations?

Overfitting



- Training error decreases with higher flexibility of the model
- We are not so much interested in the training error, but in the **generalization error**: How well does the model perform when we predict at previously unseen input locations?
- Maximum likelihood often runs into **overfitting** problems, i.e., we exploit the flexibility of the model to fit to the noise in the data

MAP Estimation

- Empirical observation: Parametric models that overfit tend to have some extreme (large amplitude) parameter values

MAP Estimation

- Empirical observation: Parametric models that overfit tend to have some extreme (large amplitude) parameter values
- Mitigate the effect of overfitting by placing a prior distribution $p(\boldsymbol{\theta})$ on the parameters
  - Penalize extreme values that are implausible under that prior

- Empirical observation: Parametric models that overfit tend to have some extreme (large amplitude) parameter values
- Mitigate the effect of overfitting by placing a prior distribution $p(\boldsymbol{\theta})$ on the parameters
  - Penalize extreme values that are implausible under that prior
- Choose $\boldsymbol{\theta}^*$ as the parameter that maximizes the (log) parameter posterior

$$\log p(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{y}) = \underbrace{\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})}_{\text{log-likelihood}} + \underbrace{\log p(\boldsymbol{\theta})}_{\text{log-prior}} + \text{const}$$

MAP Estimation

- Empirical observation: Parametric models that overfit tend to have some extreme (large amplitude) parameter values
- Mitigate the effect of overfitting by placing a prior distribution $p(\boldsymbol{\theta})$ on the parameters
  - ▶▶ Penalize extreme values that are implausible under that prior
- Choose $\boldsymbol{\theta}^*$ as the parameter that maximizes the (log) parameter posterior

$$\log p(\boldsymbol{\theta}|\boldsymbol{X},\boldsymbol{y}) = \underbrace{\log p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta})}_{\text{log-likelihood}} + \underbrace{\log p(\boldsymbol{\theta})}_{\text{log-prior}} + \text{const}$$

- Log-prior induces a direct penalty on the parameters

MAP Estimation

- Empirical observation: Parametric models that overfit tend to have some extreme (large amplitude) parameter values
- Mitigate the effect of overfitting by placing a prior distribution $p(\boldsymbol{\theta})$ on the parameters
  - Penalize extreme values that are implausible under that prior
- Choose $\boldsymbol{\theta}^*$ as the parameter that maximizes the (log) parameter posterior

$$\log p(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{y}) = \underbrace{\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})}_{\text{log-likelihood}} + \underbrace{\log p(\boldsymbol{\theta})}_{\text{log-prior}} + \text{const}$$

- Log-prior induces a direct penalty on the parameters
- **Maximum a posteriori estimate** (regularized least squares)

- Gaussian parameter prior $p(\boldsymbol{\theta}) = \mathcal{N}\big(\mathbf{0},\, \alpha^2 \boldsymbol{I}\big)$
- Log-posterior distribution:

$$
\begin{aligned}
\log p(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{y}) &= -\tfrac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}) - \tfrac{1}{2\alpha^2}\boldsymbol{\theta}^\top\boldsymbol{\theta} + \text{const} \\
&= -\tfrac{1}{2\sigma^2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\|^2 - \tfrac{1}{2\alpha^2}\|\boldsymbol{\theta}\|^2 + \text{const}
\end{aligned}
$$

MAP Estimation (2)

- Gaussian parameter prior $p(\boldsymbol{\theta}) = \mathcal{N}\big(\mathbf{0},\, \alpha^2 \boldsymbol{I}\big)$
- Log-posterior distribution:

$$
\begin{aligned}
\log p(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{y}) &= -\frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}) - \frac{1}{2\alpha^2}\boldsymbol{\theta}^\top\boldsymbol{\theta} + \text{const} \\
&= -\frac{1}{2\sigma^2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\|^2 - \frac{1}{2\alpha^2}\|\boldsymbol{\theta}\|^2 + \text{const}
\end{aligned}
$$

- Compute gradient with respect to $\boldsymbol{\theta}$, set it to $\mathbf{0}$

  Maximum a posteriori estimate:

$$
\boldsymbol{\theta}^{\text{MAP}} = (\boldsymbol{X}^\top\boldsymbol{X} + \frac{\sigma^2}{\alpha^2}\boldsymbol{I})^{-1}\boldsymbol{X}^\top\boldsymbol{y}
$$

Example: Polynomial Regression

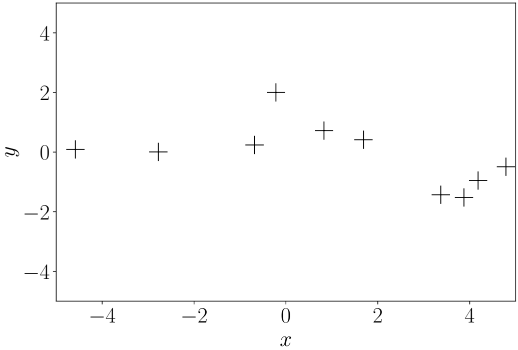


Figure: Training data

Mean prediction:

$$\mathbb{E}[y_*|\boldsymbol{x}_*, \boldsymbol{\theta}^{\text{MAP}}] = \boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{\theta}^{\text{MAP}}$$

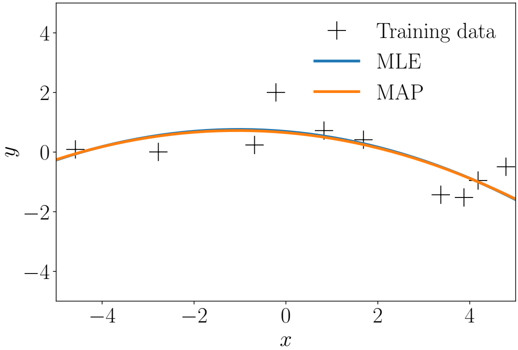


Figure: 2nd-order polynomial

Mean prediction:

$$\mathbb{E}[y_* | \boldsymbol{x}_*, \boldsymbol{\theta}^{\text{MAP}}] = \boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{\theta}^{\text{MAP}}$$

Example: Polynomial Regression

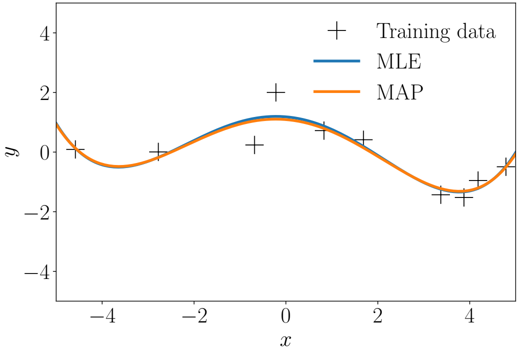


Figure: 4th-order polynomial

Mean prediction:

$$\mathbb{E}[y_*|\boldsymbol{x}_*, \boldsymbol{\theta}^{\mathsf{MAP}}] = \boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{\theta}^{\mathsf{MAP}}$$

Example: Polynomial Regression

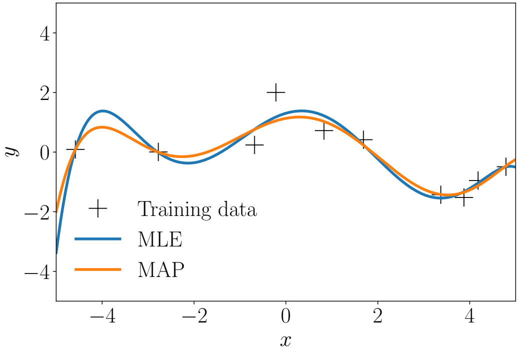


Figure: 6th-order polynomial

Mean prediction:

$$\mathbb{E}[y_*|\boldsymbol{x}_*, \boldsymbol{\theta}^{\text{MAP}}] = \boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{\theta}^{\text{MAP}}$$

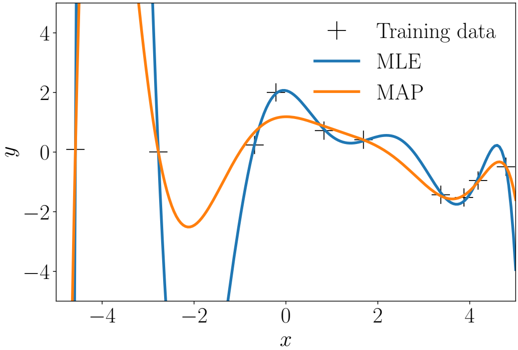


Figure: 8th-order polynomial

Mean prediction:

$$\mathbb{E}[y_*|\boldsymbol{x}_*, \boldsymbol{\theta}^{\mathsf{MAP}}] = \boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{\theta}^{\mathsf{MAP}}$$

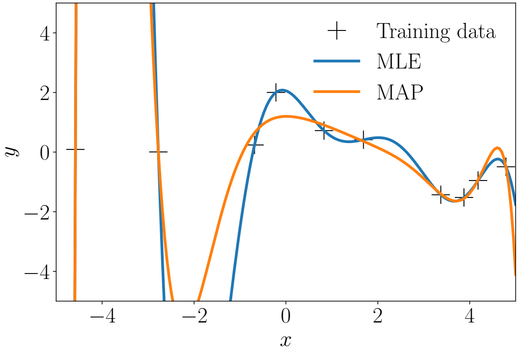
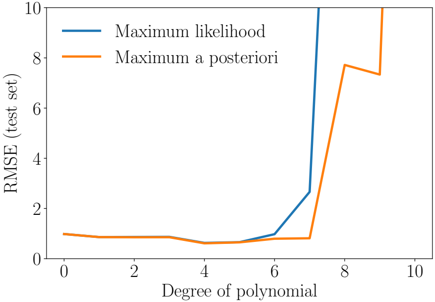


Figure: 10th-order polynomial

Mean prediction:

$$\mathbb{E}[y_*|\boldsymbol{x}_*, \boldsymbol{\theta}^{\text{MAP}}] = \boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{\theta}^{\text{MAP}}$$

Generalization Error



- MAP estimation "delays" the problem of overfitting
- It does not provide a general solution

▶▶ Need a more principled solution

Bayesian Linear Regression

$$y = \boldsymbol{\phi}^\top(\boldsymbol{x})\boldsymbol{\theta} + \epsilon\,, \quad \epsilon \sim \mathcal{N}\big(0,\, \sigma^2\big)$$

- Avoid overfitting by not fitting any parameters:
  - ▸▸ Integrate parameters out instead of optimizing them

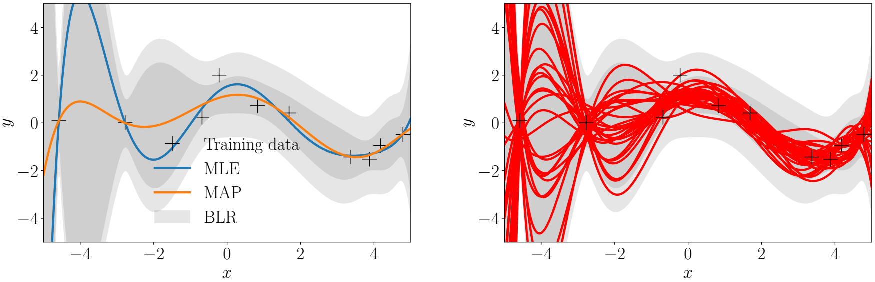
$$y = \boldsymbol{\phi}^\top(\boldsymbol{x})\boldsymbol{\theta} + \epsilon\,, \quad \epsilon \sim \mathcal{N}\big(0,\, \sigma^2\big)$$

- Avoid overfitting by not fitting any parameters:
  - Integrate parameters out instead of optimizing them
- Use a full parameter distribution $p(\boldsymbol{\theta})$ (and not a single point estimate $\boldsymbol{\theta}^*$) when making predictions:

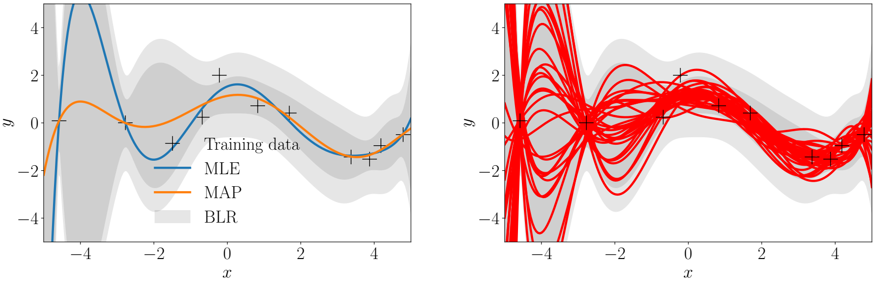
$$p(y|\boldsymbol{x}_*) = \int p(y|\boldsymbol{x}_*, \boldsymbol{\theta})p(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}$$

  - Prediction no longer depends on $\boldsymbol{\theta}$
- Predictive distribution reflects the uncertainty about the “correct” parameter setting

Example



- Light-gray: uncertainty due to noise (same as in MLE/MAP)
- Dark-gray: uncertainty due to parameter uncertainty



- Light-gray: uncertainty due to noise (same as in MLE/MAP)
- Dark-gray: uncertainty due to parameter uncertainty
- Right: Plausible functions under the parameter distribution (every single parameter setting describes one function)

Model for Bayesian Linear Regression

$$\text{Prior} \qquad p(\boldsymbol{\theta}) = \mathcal{N}\big(\boldsymbol{m}_0,\, \boldsymbol{S}_0\big)\,,$$
$$\text{Likelihood} \quad p(y|\boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}\big(y \,|\, \boldsymbol{\phi}^\top(\boldsymbol{x})\boldsymbol{\theta},\, \sigma^2\big)$$

- Parameter $\boldsymbol{\theta}$ becomes a latent (random) variable
- Prior distribution induces a distribution over plausible functions
- Choose a conjugate Gaussian prior
  - Closed-form computations
  - Gaussian posterior

Parameter Posterior and Predictions

- Prior $p(\boldsymbol{\theta}) = \mathcal{N}\big(\boldsymbol{m}_0,\, \boldsymbol{S}_0\big)$ is Gaussian ⏩ posterior is Gaussian: ⏩ **Derive this**

$$
\begin{aligned}
p(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{y}) &= \mathcal{N}\big(\boldsymbol{m}_N,\, \boldsymbol{S}_N\big) \\
\boldsymbol{S}_N &= (\boldsymbol{S}_0^{-1} + \sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1} \\
\boldsymbol{m}_N &= \boldsymbol{S}_N(\boldsymbol{S}_0^{-1}\boldsymbol{m}_0 + \sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{y})
\end{aligned}
$$

Parameter Posterior and Predictions

- Prior $p(\boldsymbol{\theta}) = \mathcal{N}\big(\boldsymbol{m}_0,\, \boldsymbol{S}_0\big)$ is Gaussian ⏩ posterior is Gaussian:

$$\begin{aligned} p(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{y}) &= \mathcal{N}\big(\boldsymbol{m}_N,\, \boldsymbol{S}_N\big) \\ \boldsymbol{S}_N &= (\boldsymbol{S}_0^{-1} + \sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1} \\ \boldsymbol{m}_N &= \boldsymbol{S}_N(\boldsymbol{S}_0^{-1}\boldsymbol{m}_0 + \sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{y}) \end{aligned}$$

- Mean $\boldsymbol{m}_N$ identical to MAP estimate

Parameter Posterior and Predictions

- Prior $p(\boldsymbol{\theta}) = \mathcal{N}\big(\boldsymbol{m}_0,\, \boldsymbol{S}_0\big)$ is Gaussian ⏩ posterior is Gaussian:

$$\begin{aligned} p(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{y}) &= \mathcal{N}\big(\boldsymbol{m}_N,\, \boldsymbol{S}_N\big) \\ \boldsymbol{S}_N &= (\boldsymbol{S}_0^{-1} + \sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1} \\ \boldsymbol{m}_N &= \boldsymbol{S}_N(\boldsymbol{S}_0^{-1}\boldsymbol{m}_0 + \sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{y}) \end{aligned}$$

- Mean $\boldsymbol{m}_N$ identical to MAP estimate
- Assume a Gaussian distribution $p(\boldsymbol{\theta}) = \mathcal{N}\big(\boldsymbol{m}_N,\, \boldsymbol{S}_N\big)$. Then

$$p(y_*|\boldsymbol{x}_*) = \mathcal{N}\big(y \,|\, \boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{m}_N,\; \boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{S}_N\boldsymbol{\phi}(\boldsymbol{x}_*) + \sigma^2\big)$$

Parameter Posterior and Predictions

- Prior $p(\boldsymbol{\theta}) = \mathcal{N}\big(\boldsymbol{m}_0,\, \boldsymbol{S}_0\big)$ is Gaussian ⏩ posterior is Gaussian:

$$
\begin{aligned}
p(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{y}) &= \mathcal{N}\big(\boldsymbol{m}_N,\, \boldsymbol{S}_N\big) \\
\boldsymbol{S}_N &= (\boldsymbol{S}_0^{-1} + \sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1} \\
\boldsymbol{m}_N &= \boldsymbol{S}_N(\boldsymbol{S}_0^{-1}\boldsymbol{m}_0 + \sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{y})
\end{aligned}
$$

- Mean $\boldsymbol{m}_N$ identical to MAP estimate
- Assume a Gaussian distribution $p(\boldsymbol{\theta}) = \mathcal{N}\big(\boldsymbol{m}_N,\, \boldsymbol{S}_N\big)$. Then

$$
p(y_*|\boldsymbol{x}_*) = \mathcal{N}\big(y\,|\,\boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{m}_N,\; \boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{S}_N\boldsymbol{\phi}(\boldsymbol{x}_*) + \sigma^2\big)
$$

- $\boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{S}_N\boldsymbol{\phi}(\boldsymbol{x}_*)$: Accounts for parameter uncertainty in predictive variance

More details ⏩ `https://mml-book.com`, Chapter 9

Marginal Likelihood

- Marginal likelihood can be computed analytically.
- With $p(\boldsymbol{\theta}) = \mathcal{N}\big(\boldsymbol{\mu},\, \boldsymbol{\Sigma}\big)$

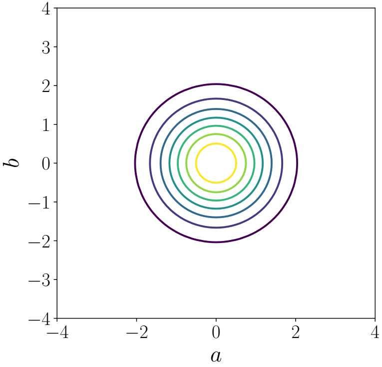
$$p(\boldsymbol{y}|\boldsymbol{X}) = \int p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta})p(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} = \mathcal{N}\big(\boldsymbol{y} \,|\, \boldsymbol{\Phi}\boldsymbol{\mu},\, \boldsymbol{\Phi}\boldsymbol{\Sigma}\boldsymbol{\Phi}^\top + \sigma^2\boldsymbol{I}\big)$$

- Derivation via completing the squares (see Section 9.3.5 of MML book)

Distribution over Functions

Consider a linear regression setting

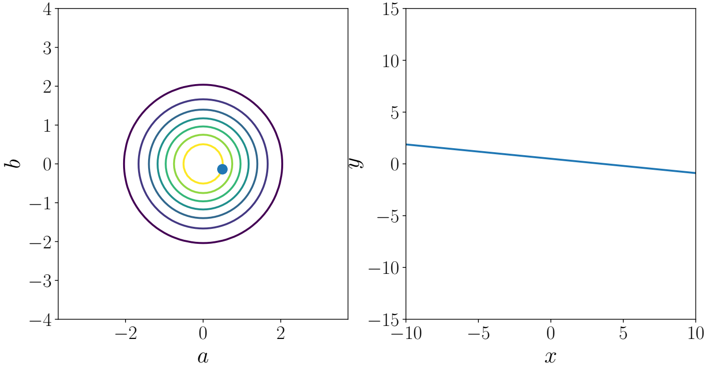
$$y = f(x) + \epsilon = a + bx + \epsilon\,, \quad \epsilon \sim \mathcal{N}\big(0,\, \sigma_n^2\big)$$
$$p(a, b) = \mathcal{N}\big(\mathbf{0},\, \boldsymbol{I}\big)$$



Sampling from the Prior over Functions

Consider a linear regression setting

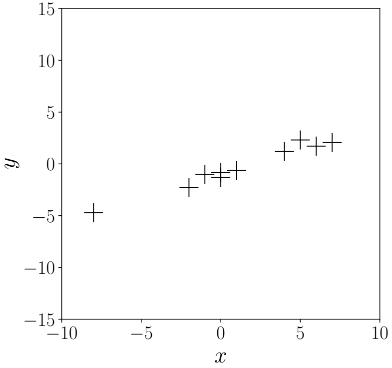
$$y = f(x) + \epsilon = a + bx + \epsilon\,, \quad \epsilon \sim \mathcal{N}\big(0,\, \sigma_n^2\big)$$
$$p(a, b) = \mathcal{N}\big(\mathbf{0},\, \boldsymbol{I}\big)$$
$$f_i(x) = a_i + b_i x, \quad [a_i, b_i] \sim p(a, b)$$



Sampling from the Posterior over Functions

Consider a linear regression setting

$$y = f(x) + \epsilon = a + bx + \epsilon\,, \quad \epsilon \sim \mathcal{N}\big(0,\, \sigma_n^2\big)$$
$$p(a, b) = \mathcal{N}\big(\mathbf{0},\, \boldsymbol{I}\big)$$
$$\boldsymbol{X} = [x_1, \dots, x_N],\ \boldsymbol{y} = [y_1, \dots, y_N] \quad \text{Training inputs/targets}$$

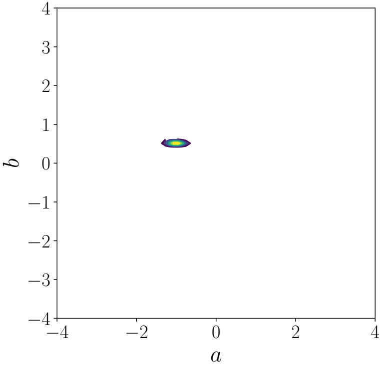


Sampling from the Posterior over Functions

Consider a linear regression setting

$$y = f(x) + \epsilon = a + bx + \epsilon\,, \quad \epsilon \sim \mathcal{N}\big(0,\, \sigma_n^2\big)$$
$$p(a, b) = \mathcal{N}\big(\mathbf{0},\, \boldsymbol{I}\big)$$
$$p(a, b|\boldsymbol{X}, \boldsymbol{y}) = \mathcal{N}\big(\boldsymbol{m}_N,\, \boldsymbol{S}_N\big)$$

Posterior



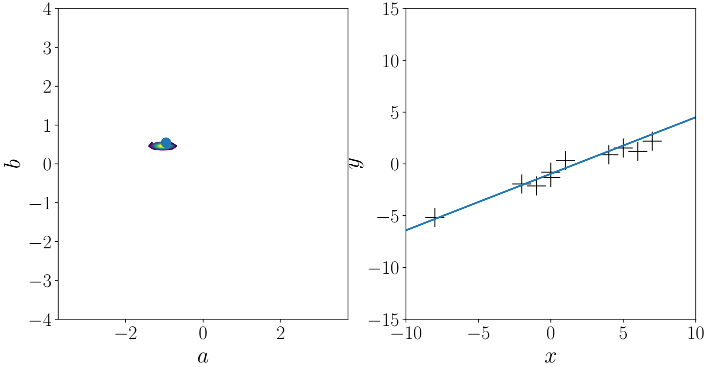
Sampling from the Posterior over Functions

Consider a linear regression setting

$$y = f(x) + \epsilon = a + bx + \epsilon\,, \quad \epsilon \sim \mathcal{N}\big(0,\, \sigma_n^2\big)$$

$$[a_i, b_i] \sim p(a, b|\boldsymbol{X}, \boldsymbol{y})$$

$$f_i = a_i + b_i x$$



Fitting Nonlinear Functions

- Fit nonlinear functions using (Bayesian) linear regression:
  Linear combination of nonlinear features

- Fit nonlinear functions using (Bayesian) linear regression:
  Linear combination of nonlinear features
- Example: Radial-basis-function (RBF) network

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \theta_i \phi_i(\boldsymbol{x})\,, \quad \theta_i \sim \mathcal{N}\big(0,\, \sigma_p^2\big)$$

Fitting Nonlinear Functions

- Fit nonlinear functions using (Bayesian) linear regression: Linear combination of nonlinear features
- Example: Radial-basis-function (RBF) network

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \theta_i \phi_i(\boldsymbol{x})\,, \quad \theta_i \sim \mathcal{N}\big(0,\, \sigma_p^2\big)$$

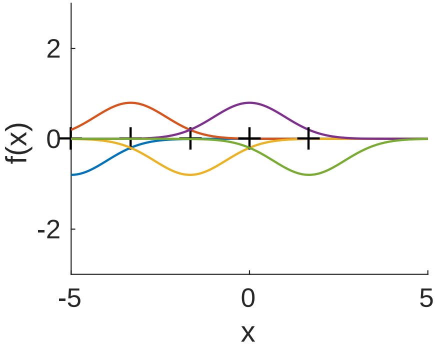
where

$$\phi_i(\boldsymbol{x}) = \exp\big(-\tfrac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^\top(\boldsymbol{x}-\boldsymbol{\mu}_i)\big)$$

for given "centers" $\boldsymbol{\mu}_i$

Illustration: Fitting a Radial Basis Function Network

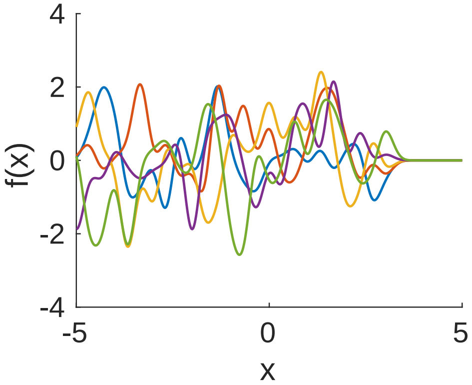
$$\phi_i(\boldsymbol{x}) = \exp\left(-\tfrac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^\top(\boldsymbol{x}-\boldsymbol{\mu}_i)\right)$$



- Place Gaussian-shaped basis functions ϕ_i at 25 input locations μ_i, linearly spaced in the interval $[-5, 3]$

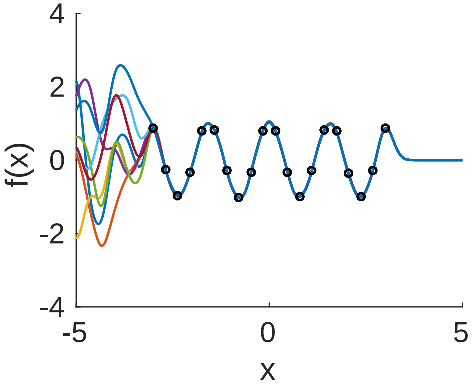
Samples from the RBF Prior

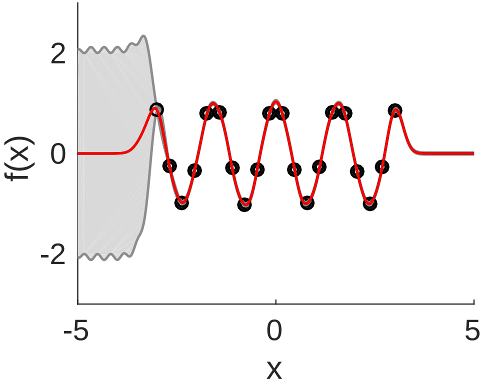
$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \theta_i \phi_i(\boldsymbol{x}), \quad p(\boldsymbol{\theta}) = \mathcal{N}\big(\mathbf{0},\, \boldsymbol{I}\big)$$



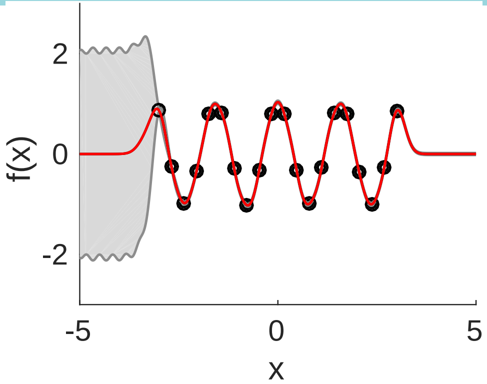
Samples from the RBF Posterior

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \theta_i \phi_i(\boldsymbol{x}), \quad p(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{y}) = \mathcal{N}\big(\boldsymbol{m}_N, \boldsymbol{S}_N\big)$$




2
0
-2
f(x)
-5
0
5
x

Limitations



- Feature engineering (what basis functions to use?)
- Finite number of features:
  - Above: Without basis functions on the right, we cannot express any variability of the function
  - Ideally: Add more (infinitely many) basis functions

Approach

- Instead of sampling parameters, which induce a distribution over functions, sample functions directly
  - Place a prior on functions
  - Make assumptions on the distribution of functions

Approach

- Instead of sampling parameters, which induce a distribution over functions, sample functions directly
  - Place a prior on functions
  - Make assumptions on the distribution of functions
- Intuition: function = infinitely long vector of function values
  - Make assumptions on the distribution of function values

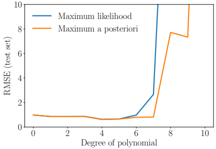
Approach

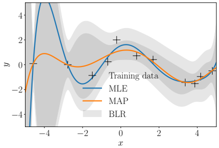
- Instead of sampling parameters, which induce a distribution over functions, sample functions directly
  - Place a prior on functions
  - Make assumptions on the distribution of functions
- Intuition: function = infinitely long vector of function values
  - Make assumptions on the distribution of function values

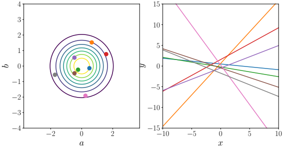
Approach

- Instead of sampling parameters, which induce a distribution over functions, sample functions directly
  - Place a prior on functions
  - Make assumptions on the distribution of functions
- Intuition: function = infinitely long vector of function values
  - Make assumptions on the distribution of function values

Gaussian process







- Regression = curve fitting
- Linear regression = linear in the parameters
- Parameter estimation via maximum likelihood and MAP estimation can lead to overfitting
- Bayesian linear regression addresses this issue, but may not be analytically tractable
- Predictive uncertainty in Bayesian linear regression explicitly accounts for parameter uncertainty
- Distribution over parameters ⏩ Distribution over functions

Appendix

Joint Gaussian Distribution

- Joint Gaussian distribution

$$p(\boldsymbol{x}, \boldsymbol{y}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy} \end{bmatrix}\right)$$

Joint Gaussian Distribution

- Joint Gaussian distribution

$$p(\boldsymbol{x}, \boldsymbol{y}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy} \end{bmatrix}\right)$$

- Marginal:

$$\begin{aligned} p(\boldsymbol{x}) &= \int p(\boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{y} \\ &= \mathcal{N}\big(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_{xx}\big) \end{aligned}$$

- Joint Gaussian distribution

$$p(\boldsymbol{x}, \boldsymbol{y}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy} \end{bmatrix}\right)$$

- Marginal:

$$\begin{aligned} p(\boldsymbol{x}) &= \int p(\boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{y} \\ &= \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_{xx}) \end{aligned}$$

- Conditional:

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{y}) &= \mathcal{N}(\boldsymbol{\mu}_{x|y}, \boldsymbol{\Sigma}_{x|y}) \\ \boldsymbol{\mu}_{x|y} &= \boldsymbol{\mu}_x + \boldsymbol{\Sigma}_{xy} \boldsymbol{\Sigma}_{yy}^{-1} (\boldsymbol{y} - \boldsymbol{\mu}_y) \\ \boldsymbol{\Sigma}_{x|y} &= \boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{xy} \boldsymbol{\Sigma}_{yy}^{-1} \boldsymbol{\Sigma}_{yx} \end{aligned}$$

If $\boldsymbol{x} \sim \mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{\mu},\, \boldsymbol{\Sigma}\big)$ and $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$ then

$$p(\boldsymbol{z}) = \mathcal{N}\big(\boldsymbol{z} \,|\, \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b},\, \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^\top\big)$$

Product of Two Gaussians

$\boldsymbol{x} \in \mathbb{R}^D$. Then:

$$\mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{a},\, \boldsymbol{A}\big)\mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{b},\, \boldsymbol{B}\big) = Z\mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{c},\, \boldsymbol{C}\big)$$

$$\boldsymbol{C} = (\boldsymbol{A}^{-1} + \boldsymbol{B}^{-1})^{-1}$$

$$\boldsymbol{c} = \boldsymbol{C}(\boldsymbol{A}^{-1}\boldsymbol{a} + \boldsymbol{B}^{-1}\boldsymbol{b})$$

$$Z = (2\pi)^{-\frac{D}{2}}|\boldsymbol{A} + \boldsymbol{B}| \exp\big(-\tfrac{1}{2}(\boldsymbol{a} - \boldsymbol{b})^\top(\boldsymbol{A} + \boldsymbol{B})^{-1}(\boldsymbol{a} - \boldsymbol{b})\big)$$

Product of Two Gaussians

$\boldsymbol{x} \in \mathbb{R}^D$. Then:

$$
\begin{aligned}
&\mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{a},\, \boldsymbol{A}\big)\mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{b},\, \boldsymbol{B}\big) = Z\mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{c},\, \boldsymbol{C}\big)\\
&\boldsymbol{C} = (\boldsymbol{A}^{-1} + \boldsymbol{B}^{-1})^{-1}\\
&\boldsymbol{c} = \boldsymbol{C}(\boldsymbol{A}^{-1}\boldsymbol{a} + \boldsymbol{B}^{-1}\boldsymbol{b})\\
&Z = (2\pi)^{-\frac{D}{2}}|\boldsymbol{A} + \boldsymbol{B}| \exp\big(-\tfrac{1}{2}(\boldsymbol{a} - \boldsymbol{b})^\top(\boldsymbol{A} + \boldsymbol{B})^{-1}(\boldsymbol{a} - \boldsymbol{b})\big)
\end{aligned}
$$

- Product of two Gaussians is an unnormalized Gaussian

Product of Two Gaussians

$\boldsymbol{x} \in \mathbb{R}^D$. Then:

$$
\begin{aligned}
&\mathcal{N}\big(\boldsymbol{x}\,|\,\boldsymbol{a},\,\boldsymbol{A}\big)\mathcal{N}\big(\boldsymbol{x}\,|\,\boldsymbol{b},\,\boldsymbol{B}\big) = Z\mathcal{N}\big(\boldsymbol{x}\,|\,\boldsymbol{c},\,\boldsymbol{C}\big)\\
&\boldsymbol{C} = (\boldsymbol{A}^{-1} + \boldsymbol{B}^{-1})^{-1}\\
&\boldsymbol{c} = \boldsymbol{C}(\boldsymbol{A}^{-1}\boldsymbol{a} + \boldsymbol{B}^{-1}\boldsymbol{b})\\
&Z = (2\pi)^{-\frac{D}{2}}|\boldsymbol{A} + \boldsymbol{B}|\exp\big(-\tfrac{1}{2}(\boldsymbol{a} - \boldsymbol{b})^\top(\boldsymbol{A} + \boldsymbol{B})^{-1}(\boldsymbol{a} - \boldsymbol{b})\big)
\end{aligned}
$$

- Product of two Gaussians is an unnormalized Gaussian
- The “un-normalizer” Z has a Gaussian functional form:

$$Z = \mathcal{N}\big(\boldsymbol{a}\,|\,\boldsymbol{b},\,\boldsymbol{A} + \boldsymbol{B}\big) = \mathcal{N}\big(\boldsymbol{b}\,|\,\boldsymbol{a},\,\boldsymbol{A} + \boldsymbol{B}\big)$$

Note: This is not a distribution (no random variables)

Example: Marginalization of a Product

$$
\begin{aligned}
p_1(\boldsymbol{x}) &= \mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{a},\ \boldsymbol{A}\big) \\
p_2(\boldsymbol{x}) &= \mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{b},\ \boldsymbol{B}\big)
\end{aligned}
$$

Then

$$
\int p_1(\boldsymbol{x}) p_2(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \qquad\qquad\qquad\qquad \in \mathbb{R}
$$

Note: In this context, $\mathcal{N}$ is used to describe the functional relationship between $\boldsymbol{a}, \boldsymbol{b}$. Do not treat $\boldsymbol{a}$ or $\boldsymbol{b}$ as random variables—they are both deterministic quantities.

Example: Marginalization of a Product

$$p_1(\boldsymbol{x}) = \mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{a},\ \boldsymbol{A}\big)$$
$$p_2(\boldsymbol{x}) = \mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{b},\ \boldsymbol{B}\big)$$

Then

$$\int p_1(\boldsymbol{x}) p_2(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = Z = \mathcal{N}\big(\boldsymbol{a} \,|\, \boldsymbol{b},\ \boldsymbol{A} + \boldsymbol{B}\big) \in \mathbb{R}$$

Note: In this context, $\mathcal{N}$ is used to describe the functional relationship between $\boldsymbol{a}, \boldsymbol{b}$. Do not treat $\boldsymbol{a}$ or $\boldsymbol{b}$ as random variables—they are both deterministic quantities.